Package ‘grpCox’

September 16, 2020

Type Package
Title Penalized Cox Model for High-Dimensional Data with Grouped Predictors
Version 1.0.1
Date 2020-09-07
Author Xuan Dang
Maintainer Xuan Dang <xuandang11289@gmail.com>
Description Fit the penalized Cox models with both non-overlapping and overlapping grouped penalties including the group lasso, group smoothly clipped absolute deviation, and group mini-max concave penalty. The algorithms combine the MM approach and group-wise descent with some computational tricks including the screening, active set, and warm-start. Different tuning regularization parameter methods are provided.
License GPL (>= 2)
Encoding UTF-8
Imports Rcpp (>= 1.0.3)
LinkingTo Rcpp, RcppEigen
Depends Matrix (>= 1.2-10), MASS, colorspace
RoxygenNote 7.0.2
NeedsCompilation yes
Repository CRAN
Date/Publication 2020-09-16 09:00:03 UTC

R topics documented:

grpCox-package ................................................................. 2
cv.grpCox ................................................................. 3
cv.grpCoxOverlap .......................................................... 5
grpCox ................................................................. 7
grpCoxOverlap ........................................................... 9
plot.Coef ............................................................... 11
plot.gCoef .............................................................. 12
plot.llCV .............................................................. 14
Penalized Cox Model for High-Dimensional Data with Grouped Predictors

Description

Fit the penalized Cox models with both non-overlapping and overlapping grouped penalties including the group lasso, group smoothly clipped absolute deviation, and group minimax concave penalty. The algorithms combine the MM approach and group-wise descent with some computational tricks including the screening, active set, and warm-start. Different tuning regularization parameter methods are provided.

Package Content

Index of help topics:

- cv.grpCox: Cross-validation for grpCox
- cv.grpCoxOverlap: Cross-validation for grpCoxOverlap
- grpCox: Fit a penalized Cox model.
- grpCox-package: Penalized Cox Model for High-Dimensional Data with Grouped Predictors
-grpCoxOverlap: Fit a penalized regression path with overlapping grouped covariates.
- plot.Coe: Plots the coefficient paths
- plot.gCoe: Plots the coefficient paths with the same color for the covariates in the same group.
- plot.llCV: Plot the cross-validation curve produced by cv.grpCox or cv.grpCoxOverlap

Maintainer

Xuan Dang <xuandang11289@gmail.com>

Author(s)

Xuan Dang
cv.grpCox

Cross-validation for grpCox

Description

Does k-fold cross-validation for grpCox

Usage

cv.grpCox(X, y, g, m, penalty=c("glasso", "gSCAD", "gMCP"), lambda=NULL, nlambda=100, rlambda=NULL, gamma=switch(penalty, SCAD = 3.7, 3), standardize=TRUE, thresh=1e-3, maxit=1e+4, nfolds=10, foldid=NULL)

Arguments

X The design matrix.

y The response vector includes time corresponding to failure/censor times, and status indicating failure (1) or censoring (0).

g A vector indicating the group structure of the covariates. It can be unordered groups.

m Group multipliers. Default is the square root of group size.

penalty The penalty to be applied to the model. It is one of glasso, gSCAD, or gMCP.

lambda A user supplied sequence of lambda values. If it is left unspecified, and the function automatically computes a grid of lambda values.

nlambda The number of lambda values to use in the regularization path. Default is 100.

rlambda Smallest value for lambda, as a fraction of the maximum lambda, the (data derived) entry value (i.e. the smallest value for which all coefficients are zero). The default depends on the sample size relative to the number of covariates. If sample size>#covariates, the default is 0.001, close to zero. If sample size>#covariates, the default is 0.05.

gamma Tuning parameter of the group SCAD/MCP penalty. Default is 3.7 for SCAD and 3 for MCP.

standardize Logical flag for variable standardization prior to fitting the model.

thresh Convergence threshold for one-step coordinate descent. Defaults value is 1E-7.

maxit Maximum number of passes over the data for all lambda values; default is 1E+5.

nfolds The number of cross-validation folds. Default is 10.

foldid An optional vector of values between 1 and nfolds identifying what fold each observation is in.
Value

- **aBetaSTD**: A standardized coefficient matrix whose columns correspond to nlambda values of lambda.
- **aBetaO**: A coefficient matrix (without standardization) whose columns correspond to nlambda values of lambda.
- **mBetaSTD**: The coefficient in standardized form gives maximum log-likelihood value using the first cross-validation method.
- **mBetaO**: The coefficient in original form gives maximum log-likelihood value using the first cross-validation method.
- **pBetaSTD**: The coefficient in standardized form gives maximum log-likelihood value using the penalized cross-validation method.
- **pBetaO**: The coefficient in original form gives maximum log-likelihood value using the penalized cross-validation method.
- **fit**: A matrix includes lambda value, the mean cross-validation error.
- **lambda**: The lambda values used.
- **g**: A vector indicating the group structure of the covariates.
- **cvm**: The maximum value of log likelihood.
- **lambda.max**: The value of lambda corresponds to the maximum value of log likelihood using the first cross-validation method.
- **lambda.pcvl**: The value of lambda corresponds to the maximum value of log likelihood using the penalized cross-validation method.

Author(s)

Xuan Dang <<xuandang11289@gmail.com>>

References


Examples

```r
set.seed(200)
N <- 50
p <- 9
x <- matrix(rnorm(N * p), nrow = N)
beta <- c(.65,.65,0,0,.65,.65,0,.65,0)
hx <- exp(x %*% beta)
ty <- rexp(N,hx)
tcens <- 1 - rbinom(n=N, prob = 0.2, size = 1)
y <- data.frame(illt=ty, ills=tcens)
names(y) <- c("time", "status")
```
cv.grpCoxOverlap

Describes s-fold cross-validation for grpCoxOverlap

Usage

cv.grpCoxOverlap(X0, y, group, penalty=c("glasso", "gSCAD", "gMCP"), lambda=NULL, nlambdas=100, rlambda=NULL, gamma=switch(penalty, SCAD = 3.7, 3), standardize=TRUE, thresh=1e-3, maxit=1e+4, nflows=10, foldid=NULL, returnLatent=TRUE)

Arguments

X0 The design matrix.
y The response vector includes time corresponding to failure/censor times, and status indicating failure (1) or censoring (0).
group A list of groups, each includes indices of covariates in the group.
penalty The penalty to be applied to the model. It is one of "glasso", "gSCAD", or "gMCP".
lambda A user supplied sequence of lambda values. If it is left unspecified, the function automatically computes a grid of lambda values.
rlambda The number of lambda values to use in the regularization path. Default is 100.
rlambda Smallest value for lambda, as a fraction of the maximum lambda, the (data derived) entry value (i.e. the smallest value for which all coefficients are zero). The default depends on the sample size relative to the number of covariates. If sample size > #covariates, the default is 0.001, close to zero. If sample size > #covariates, the default is 0.05.
gamma Tuning parameter of the group SCAD/MCP penalty. Default is 3.7 for SCAD and 3 for MCP.
standardize Logical flag for variable standardization prior to fitting the model.
thresh Convergence threshold for one-step coordinate descent. Defaults value is 1E-7.
maxit Maximum number of passes over the data for all lambda values; default is 1E+5.
nflows The number of cross-validation folds. Default is 10.
foldid An optional vector of values between 1 and nflows identifying what fold each observation is in.
returnLatent Return the coefficient matrix in latent space. Default is TRUE.
Value

**A.** A coefficient matrix whose columns correspond to nlambda values of lambda in latent space.

**B.** A coefficient matrix whose columns correspond to nlambda values of lambda in original space.

**C.** The coefficient in latent space gives maximum log-likelihood value using the first cross-validation method.

**D.** The coefficient in original space gives maximum log-likelihood value using the first cross-validation method.

**E.** The coefficient in latent space gives maximum log-likelihood value using the penalized cross-validation method.

**F.** The coefficient in original space gives maximum log-likelihood value using the penalized cross-validation method.

**G.** A matrix includes lambda value, the mean cross-validation error.

**H.** The lambda values used.

**I.** A list of groups, each includes indices of covariates in the group.

**J.** A vector indicating the group structure of the covariates in latent space.

**K.** The maximum value of log likelihood.

**L.** The value of lambda corresponds to the maximum value of log likelihood using the first cross-validation method.

**M.** The value of lambda corresponds to the maximum value of log likelihood using the penalized cross-validation method.

Author(s)

Xuan Dang <<xuandang11289@gmail.com>>

References


Examples

```r
set.seed(100001)
N <- 50
p <- 6
times <- 1:p
rho <- 0.5
H <- abs(outer(times, times, "-"))
C <- 1 * rho^H
C[cbind(1:p, 1:p)] <- C[cbind(1:p, 1:p)]
sigma <- matrix(C,p,p)
```
mu <- rep(0,p)
mu <- mvrnorm(n=N, mu, sigma)

beta <- c(0,.8, 1, 2, 1, 0)
beta <- exp(x %*% beta)

hx <- exp(x %*% beta)

hx <- rexp(N,hx)

hx <- rexp(N,hx)
tcens <- 1 - rbinom(n=N, prob = 0.2, size = 1)
y <- data.frame(illt=ty, ills=tcens)
names(y) <- c("time", "status")

group <- list(g1 = c(1,2,3,4), g2 = c(1,2,6), g3 = c(2,3),
g4 = c(4,5), g5 = c(5))

cvfit <- cv.grpCoxOverlap(x, y, group, penalty="glasso", nlambda=50)
plot.llCV(cvfit)

---

**grpCox**

*Fit a penalized Cox model.*

**Description**

Fit the regularization paths for Cox models with grouped covariates.

**Usage**

```r
grpCox(X, y, g, m, penalty=c("glasso", "gSCAD", "gMCP"), lambda=NULL,
nlambda=100, rlambda=NULL, gamma=switch(penalty, gSCAD = 3.7, 3),
standardize=TRUE, thresh=1e-3, maxit=1e+4)
```

**Arguments**

- **X**  
The design matrix.
- **y**  
The response vector includes time corresponding to failure/censor times, and status indicating failure (1) or censoring (0).
- **g**  
A vector indicating the group structure of the covariates. It can be unordered groups.
- **m**  
Group multipliers. Default is the square root of group size.
- **penalty**  
The penalty to be applied to the model. It is one of glasso, gSCAD, or gMCP.
- **lambda**  
A user supplied sequence of lambda values. If it is left unspecified, and the function automatically computes a grid of lambda values.
- **nlambda**  
The number of lambda values to use in the regularization path. Default is 100.
- **rlambda**  
Smallest value for lambda, as a fraction of the maximum lambda, the (data derived) entry value (i.e. the smallest value for which all coefficients are zero). The default depends on the sample size relative to the number of covariates. If sample size>\#covariates, the default is 0.001, close to zero. If sample size>\#covariates, the default is 0.05.
gamma Tuning parameter of the group SCAD/MCP penalty. Default is 3.7 for SCAD and 3 for MCP.

standardize Logical flag for variable standardization prior to fitting the model.

thresh Convergence threshold for one-step coordinate descent. Defaults value is 1E-7.

maxit Maximum number of passes over the data for all lambda values; default is 1E+5.

Details
The group SCAD (gSCAD) and group MCP (gMCP) formulations have been presented in Wang et. al 2007, Huang et. al 2012.

Value

aBetaSTD A standardized coefficient matrix whose columns correspond to nlambda values of lambda.

aBeta0 A coefficient matrix (without standardization) whose columns correspond to nlambda values of lambda.

lambda The lambda values used.

ll The log likelihood values.

g A vector indicating the group structure of the covariates.

Author(s)
Xuan Dang <<xuandang11289@gmail.com>>

References

Examples
set.seed(200)
N <- 50
p <- 9
x <- matrix(rnorm(N * p), nrow = N)
beta <- c(0.65, 0.65, 0, 0, 0.65, 0.65, 0, 0.65, 0)
hx <- exp(x %*% beta)
ty <- rexp(N, hx)
tcens <- 1 - rbinom(n=N, prob = 0.2, size = 1)
y <- data.frame(illt=ty, ills=tcens)
names(y) <- c("time", "status")
g <- c(1,1,2,2,3,3,2,3,2)
m <- c(sqrt(2),sqrt(4),sqrt(3))
fit <- grpCox(x,y,g,m,penalty="glasso")
plot.gCoef(fit$aBetaO, fit$g, fit$lambda)

.grpCoxOverlap

**Fit a penalized regression path with overlapping grouped covariates.**

**Description**

Fit the regularization paths for Cox’s models with overlapping grouped covariates.

**Usage**

```r
grpCoxOverlap(X0, y, group, penalty=c("glasso", "gSCAD", "gMCP"),
lambda=NULL, nlambda=100, rlambda=NULL, gamma=switch(penalty, gSCAD = 3.7, 3),
standardize = TRUE, thresh=1e-3, maxit=1e+4, returnLatent=TRUE)
```

**Arguments**

- `X0` The design matrix.
- `y` The response vector includes time corresponding to failure/censor times, and status indicating failure (1) or censoring (0).
- `group` A list of groups, each includes indices of covariates in the group.
- `penalty` The penalty to be applied to the model. It is one of glasso, gSCAD, or gMCP.
- `lambda` A user supplied sequence of lambda values. If it is left unspecified, the function automatically computes a grid of lambda values.
- `nlambda` The number of lambda values to use in the regularization path. Default is 100.
- `rlambda` Smallest value for lambda, as a fraction of the maximum lambda, the (data derived) entry value (i.e. the smallest value for which all coefficients are zero). The default depends on the sample size relative to the number of covariates. If sample size>#covariates, the default is 0.001, close to zero. If sample size>#covariates, the default is 0.05.
- `gamma` Tuning parameter of the group SCAD/MCP penalty. Default is 3.7 for SCAD and 3 for MCP.
- `standardize` Logical flag for variable standardization prior to fitting the model.
- `thresh` Convergence threshold for one-step coordinate descent. Defaults value is 1E-7.
- `maxit` Maximum number of passes over the data for all lambda values; default is 1E+5.
- `returnLatent` Return the coefficient matrix in latent space. Default is TRUE.

**Details**

The the group SCAD (gSCAD) and group MCP (gMCP) formulations have been presented in Wang et. al 2007, Huang et. al 2012.

The method based on the latent group approach (Jacob et al. 2009, Obozinski et al. 2011.)
Value

- **aBetaLatent**: A coefficient matrix whose columns correspond to nlambda values of lambda in latent space.
- **aBetaOri**: A coefficient matrix whose columns correspond to nlambda values of lambda in original space.
- **lambda**: The lambda values used.
- **ll**: The log likelihood values.
- **group**: A list of groups, each includes indices of covariates in the group.
- **glatent**: A vector indicating the group structure of the covariates in latent space.

Author(s)

Xuan Dang <<xuandang11289@gmail.com>>

References


Examples

```r
set.seed(100001)
N <- 50
p <- 6
times <- 1:p
rho <- 0.5
H <- abs(outer(times, times, "-"))
C <- 1 + rho^H
C[cbind(1:p, 1:p)] <- C[cbind(1:p, 1:p)]
sigma <- matrix(C,p,p)
mu <- rep(0,p)
x <- mvrnorm(n=N, mu, sigma)
beta <- c(0, .8, 1, 2, 1, 0)
hx <- exp(x %*% beta)
ty <- rexp(N,hx)
tcens <- 1 - rbinom(n=N, prob = 0.2, size = 1)
y <- data.frame(illt=ty, ills=tcens)
names(y) <- c("time", "status")
group <- list(g1 = c(1,2,3,4), g2 = c(1,2,6), g3 = c(2,3), g4 = c(4,5), g5 = c(5))
fit <- grpCoxOverlap(x, y, group, penalty="glasso", nlambda=50)
```

# plot the coefficient values in latent space
plot.gCoef(fit$aBetaLatent, fit$glatent, fit$lambda)
# plot the coefficient values in original space
plot.Coef(fit$aBetaOri, fit$lambda)

---

**plot.Coef**

Plots the coefficient values as a function of the lambda values used.

### Description

Plots the coefficient values as a function of the lambda values used.

### Usage

```r
## S3 method for class 'Coef'
plot(x, lambda, label=TRUE, xlab="log(Lambda)", ylab="Coefficients", title=NULL, ...)
```

### Arguments

- `x` A matrix of coefficients.
- `lambda` The lambda values used.
- `label` The indices of covariates. Default is TRUE.
- `xlab` The name of the x-axis.
- `ylab` The name of the y-axis.
- `title` The title of the plot.
- `...` further arguments to plot

### Details

A plot is produced, and nothing is returned.

### Value

No return value.

### Author(s)

Xuan Dang <<xuandang11289@gmail.com>>
Examples

```r
set.seed(100001)
N <- 50
p <- 6
times <- 1:p
rho <- 0.5
H <- abs(outer(times, times, "-"))
C <- 1 * rho^H
C[cbind(1:p, 1:p)] <- C[cbind(1:p, 1:p)]
sigma <- matrix(C,p,p)
mu <- rep(0,p)
x <- mvrnorm(n=N, mu, sigma)

beta <- c(0, .8, 1, 2, 1, 0)
hx <- exp(x %*% beta)
ty <- rexp(N,hx)
tcens <- 1 - rbinom(n=N, prob = 0.2, size = 1)
y <- data.frame(illt=ty, ills=tcens)
names(y) <- c("time", "status")
group <- list(g1 = c(1,2,3,4), g2 = c(1,2,6), g3 = c(2,3), g4 = c(4,5), g5 = c(5))
fit <- grpCoxOverlap(x, y, group, penalty="glasso", nlambda=50)
# plot the coefficient values in latent space
plot.gCoef(fit$aBetaLatent, fit$glatent, fit$lambda)
# plot the coefficient values in original space
plot.Coef(fit$aBetaOri, fit$lambda)
```

Description

Plots the coefficient values as a function of the lambda values used. The covariates in the same group have the same color.

Usage

```r
## S3 method for class 'gCoef'
plot(x,g,lambda,label=TRUE,xlab="log(Lambda)",
    ylab="Coefficients", title=NULL,...)
```

Arguments

- `x`: A matrix of coefficients.
- `g`: A vector indicating the group structure of the covariates.
- `lambda`: The lambda values used.
- `label`: The indices of covariates. Default is TRUE.
plot.gCoef

xlab   The name of the x-axis.
ylab   The name of the y-axis.
title  The title of the plot.
...

Details
A plot is produced, and nothing is returned.

Value
No return value.

Author(s)
Xuan Dang <<xuandang11289@gmail.com>>

Examples

```r
set.seed(100001)
N <- 50
p <- 6
times <- 1:p
rho <- 0.5
H <- abs(outer(times, times, "-"))
C <- 1 * rho^H
C[cbind(1:p, 1:p)] <- C[cbind(1:p, 1:p)]
sigma <- matrix(C, p, p)
mu <- rep(0, p)
x <- mvrnorm(n=N, mu, sigma)

beta <- c(0, 0.8, 1, 2, 1, 0)
hx <- exp(x %*% beta)
ty <- rexp(N, hx)
tcens <- 1 - rbinom(n=N, prob = 0.2, size = 1)
y <- data.frame(illt=ty, ills=tcens)
names(y) <- c("time", "status")

group <- list(g1 = c(1,2,3,4), g2 = c(1,2,6), g3 = c(2,3), g4 = c(4,5), g5 = c(5))
fit <- grpCoxOverlap(x, y, group, penalty="glasso", nlambda=50)

# plot the coefficient values in latent space
plot.gCoef(fit$aBetaLatent, fit$glatent, fit$lambda)

# plot the coefficient values in original space
plot.Coef(fit$aBetaOri, fit$lambda)
```
plot.llCV

Plot the cross-validation curve produced by cv.grpCox or cv.grpCoxOverlap

Description

Plots the cross-validation curve, and upper and lower standard deviation curves, as a function of the lambda values used.

Usage

## S3 method for class 'llCV'
plot(x,...)

Arguments

- **x**  
  fitted cv.grpCox or cv.grpCoxOverlap object
- **...**  
  further arguments to plot

Details

A plot is produced, and nothing is returned.

Value

No return value.

Author(s)

Xuan Dang <xuandang11289@gmail.com>

Examples

```r
cvfit <- cv.grpCox(x,y,g,m,penalty="glasso")
plot.llCV(cvfit)
```

```r
set.seed(200)
N <- 50
p <- 9
x <- matrix(rnorm(N * p), nrow = N)
beta <- c(.65,.65,0,0,.65,.65,0,.65,0)
hx <- exp(x %*% beta)
hy <- exp(x %*% beta)
y <- data.frame(illt=ty, ills=tcens)
names(y) <- c("time", "status")
g <- c(1,1,2,2,3,3,2,3,2)
m <- c(sqrt(2),sqrt(4),sqrt(3))
cvfit <- cv.grpCox(x,y,g,m,penalty="glasso")
plot.llCV(cvfit)
```
Index

* Cox models
  cv.grpCox, 3
  cv.grpCoxOverlap, 5
  grpCox, 7
  grpCoxOverlap, 9
  plot.Coe, 11
  plot.gCoef, 12
  plot.llCV, 14

* group regularization
  cv.grpCox, 3
  cv.grpCoxOverlap, 5
  grpCox, 7
  grpCoxOverlap, 9
  plot.Coe, 11
  plot.gCoef, 12
  plot.llCV, 14

* overlapping group
  cv.grpCoxOverlap, 5
  grpCoxOverlap, 9
  plot.Coe, 11
  plot.gCoef, 12
  plot.llCV, 14

* package
  grpCox-package, 2
  cv.grpCox, 3
  cv.grpCoxOverlap, 5
  grpCox, 7
  grpCox-package, 2
  grpCoxOverlap, 9
  plot.Coe, 11
  plot.gCoef, 12
  plot.llCV, 14